ELSEVIER

Computer Methods and Programs in Biomedicine 70 (2003) 1–9

Computer Methods and Programs in Biomedicine

www.elsevier.com/locate/cmpb

Generic number systems and haplotype analysis

Jing Hua Zhao *, Pak Chung Sham

Section of Genetic Epidemiology and Biostatistics, Division of Psychological Medicine, Institute of Psychiatry, De Crespigny Park, Denmark Hill, London SE5 8AF, UK

Received 5 June 2001; received in revised form 21 September 2001; accepted 25 September 2001

Abstract

Three simple and elegant algorithms involving binary and mixed-radix numbers are presented as C subroutines and applied to gene-counting procedure. The first, a multikey radix-sorting subroutine, is used to tally individuals with similar genetic marker information. The second, a subroutine for N-ary number addition, is used to enumerate all possible phases of a heterozygote. The third, a mixed-radix number subroutine, is used to generate all haplotypes and indexing single array of haplotype frequencies. Examples exposing these algorithms are also given. The sorting algorithm entails broad application while the N-ary and mixed-radix number algorithms are very efficient for generic looping. Implementation of gene-counting using these algorithms avoids use of multilocus genotype identifier and improves its portability to other analysis. © 2002 Elsevier Science Ireland Ltd. All rights reserved.

Keywords: Radix sort; Mixed-radix number; EM algorithm; Gene-counting; Number theory

1. Introduction

Efficient algorithms bear nature of the specific problem to be tackled. One such example is the twin zygosity determination problem, where monozygotic or dizygotic twins are determined through examining number of alleles shared by descent at a set of genetic markers, and effectively implemented under ternary number system [1]. Similar method was used to enumerate patterns of multiple ancestral mutations [2].

Here, we extend the idea to N-ary and mixed-radix number system and apply it to our recently developed gene counting method [3–6], a particular form of EM algorithm for iterative estimation of coupling information between alleles of different loci, so that more efficient program is produced. Readers not familiar with genetic and computing terms are referred to Appendix A for their definitions first. These subroutines are short enough to be included and explained in details.

Haplotype frequency estimates from gene-counting method can be tested against equilibrium frequencies, which assumes random assortment of constituent loci to form haplotypes. Statistical significance is an indication of allelic association, which contains important information on evolution and disease-predisposing mechanism underlying human diseases. Since different individuals may have the same marker informa-

* Corresponding author. Tel.: +44-20-78480093; fax: +44-20-77019044.

E-mail address: j.zhao@iop.kcl.ac.uk (J.H. Zhao).

PII: S0169-2607(01)00193-6

tion, they can be collapsed according to this information before invoking the procedure. Currently this is achieved with either linked-list or search trees. In either method multilocus genotype identifiers are created and used as search keys. When multiple polymorphic loci are involved, the number of possible genotypes can increase rapidly so that the multilocus genotype identifier becomes very unwieldy. For example the biggest possible identifier for eight loci with alleles 8, 8, 13, 14, 2, 2, 13, 10 is $[8(8+1)/2][8(8+1)/2]\ldots[10(10+1)/2] = 557\,804\,847\,600$. The other problem associated with current method is the complexity in implementation of gene-counting, with many recursive and nested procedures involved.

A radix-sorting algorithm, an N-ary number and a mixed number routine are thus devised to address these limitations, which lead to a more elegant program. Their C codes are provided and illustrated with examples, followed by a brief discussion.

2. Methods

2.1. Basic gene-counting method

Let L be the number of loci observed in a sample of N diploid individuals, haplotype frequencies can be obtained from the number of haplotypes contributed by all observed genotypes divided by $2N$. However, the contribution of a multilocus genotype cannot be deduced if that genotype is heterozygous for two or more markers. If we let m, $m \leq L$, be the number of heterozygous loci in a multilocus genotype, then there are 2^{m-1} possible phases, referring to the 2^{m-1} possible pairs of haplotypes which could give rise to the observed genotype. Now denote the total coupling probabilities of all the possible phases as T, the contribution of each phase is then *counts* × coupling probability of that phase/T, where *counts* is the number of individuals associated with this genotype. This probability given current haplotype frequency estimates is furnished as E-step in the EM algorithm, while the haplotype counts are updated and divided by $2N$ as M-step. Starting from any initial haplotype frequencies this procedure is repeated until change in consecutive iterations becomes negligible. The convergence criteria can be based on L^1 norm, the absolute difference of these two haplotype frequency vectors, or L^2 norm, the Euclidean distance between them.

The aforementioned procedure can be summarised as the following algorithm.

2.1.1. Algorithm O (gene-counting using observed multilocus genotypes)

```
Initialise the haplotype frequencies to be the product of constituent allele frequencies.
  Repeat
  For each observed multilocus genotypes Do
    Obtain $m$, the number of heterozygous loci out of $L$ loci
    E-step:
    if $m = 0$ then count that haplotype twice
  else
      Enumerate all $2^{m-1}$ haplotypes from the multiply heterozygote and calculate their total coupling probabilities $T$, add to each haplotype count counts*(coupling probabilities/$T$)
  End do
  M-step:
      Obtained haplotype frequencies from haplotype counts divided by $2N$ Until haplotype frequency changes are negligible
```

At E-step, 2^m haplotypes are symmetric so that only 2^{m-1} of them are distinct. The gene-counting procedure obtained its name since the E- and M-steps only involve counting genes. Note it works on observed multilocus genotypes instead of all possible multilocus genotypes. It is clear that algorithm O does not imply the use of multilocus genotype identifier.

2.2. Data preparation

Current method of data preparation takes advantage of the one-to-one correspondence between the multilocus genotype and multilocus genotype identifier, which allows for the original data to be sorted by a single multilocus genotype identifier. Nevertheless it seems more natural to view the whole genotypic information of an individual as a mixed-radix number, so that the sorting problem can be approached using a modified version of radix sort.

Let array *k* be an array containing genotype identifiers at individuals loci, *M* be the largest radix, *head* [*M*] be the sorting bucket for radix sorting, *D* be the number of digits or marker loci, *s* be a linked-list containing data. Subroutine *S* below, motivated by [7], implements a radix sort to do the data preparation.

2.2.1. Subroutine S (mixed-radix sorting)

```
 1. list sort( list s, int j )
 2. {
 3. int i;
 4. list head[M], t;
 5. struct rec aux;
 6. extern list Last;
 7. if (s = = NULL) return(s);
 8. if (s- > next = = NULL) {
 9. Last = s;
10. return(s);
11. }
12. if (j > D) {
13. for (Last = s; Last- > next! = NULL; Last =
    Last- > next);
14. return(s);
15. }
16. for (i = 0; i < M; i + + ) head[i ] = NULL;
17. while (s! = NULL) {
18. i = s- > k[ j ];
19. t = s;
20. s = s- > next;
21. t- > next = head[i ];
22. head[i ] = t;
23. }
24. t = &aux;
25. for (i = 0; i < M; i + + ) if (head[i ]! =
    NULL) {
26. t- > next = sort( head[i ], j + 1);
27. t = Last;
28. }
29. return(aux.next);
30. }
```

Subroutine S also allows for a subset of markers in the original data set to be selected, so long as we change line 18 to be *i* = *s*- > *k*[selidx[*j*]], where *selidx* is an array holding in the marker list being selected if only a subset of markers are used.

2.3. Haplotype frequency estimation

The major operations in the gene-counting procedure are to enumerate all the possible phases of a heterozygous genotype and update the haplotype frequency estimates accordingly. These are achieved via Subroutines N and M below. Since we already know the alleles involved in a multilocus genotype, different phases can be obtained by switching alleles at a locus, which is readily obtained with the following subroutine using binary numbers. Let *radix* be the base of the *N*-ary number system, *d* be an array containing digits of such a number, Subroutine N is as follows.

2.3.1. Subroutine N (N-ary number addition)

```
int digit2(int radix, int d[], int i )
{
if(d[i ] < radix)
{
+ + d[i ];
goto ok;
}
else
{
d[i ] = 0;
+ + d[i + 1];
if(d[i + 1] < = radix) goto ok;
}
digit2(radix,d,i + 1);
ok:
return 0;
}
```

Array *d* is initialised as zero, and calls to this subroutine are made 2^{m-1} times. Each call switches to a different phase. This is one form of the ternary number routines in [1], but used for binary number. Note that Subroutine N can be slightly modified when subset of markers in the original sample is used.

Since a given problem may involve different number of alleles at different marker loci, it is necessary to hold allele numbers at all loci for haplotype construction. Subroutine M can be used to obtain the all the haplotypes. It extends Subroutine N by allowing for mixed radixes.

2.3.2. Subroutine M (mixed-radix number addition)

```
int digitm(int radix[], int d[], int i )
{
if(d[i ] < radix[i ])
{
+ +d[i ];
goto ok;
}
else
{
d[i ] = 0;
+ +d[i + 1];
if(d[i + 1] < = radix[i + 1]) goto ok;
}
digitm(radix,d,i + 1);
ok:
return 0;
}
```

The number of markers in a typical analysis may range between 2 and 30. It would be very clumsy to use haplotype frequencies as a multidimensional array, so a single array indexed by appropriate alleles is more desirable. Again haplotype is viewed as mixed-radix number, the index to haplotype array can be obtained via Horner's method in Subroutine H. Let *loci* be an array holding number of alleles at all loci, here as radixes or bases of the mixed radix number, *a* be an array holding haplotype. Subroutine H is shown as follows.

2.3.3. Subroutine H (evaluate mixed-radix number)

```
int linenum(int *loci, int *ai)
{
int loc,j;
loc = 0;
for( j = 1;j < = nloci;j + + )
if( j = = nloci) loc + = ai[ j-1];
else loc = (loc + ai[ j-1]-1)*loci[ j ];
return(loc);
}
```

The basic radix sorting procedure was documented in [7]. However, Subroutines B and M can be further exposed. We will give two simple examples in the following section. Practical examples can be found in [8,9].

3. Examples

In this section, we examine two simplest examples involving two and three biallelic loci. In the following we denote n's to be the observed genotype counts and h's to be the haplotype frequencies.

3.1. Two biallelic markers

Data involving biallelic markers can be organised into Table 1.

Cells of this table thus subscripted would be suitable for implementations in C. Contributions from each cell to haplotype frequencies are unambiguous except for doubly heterozygous cell 4 with count n_4, that two possibilities or phases 22/11 and 21/12 are to be differentiated. Assuming independence, i.e. haplotype frequencies being product of the constituent allele frequencies, the expected probabilities are $\alpha^4 = h_{21}h_{12}/(h_{21}h_{12} + h_{22}h_{11}$ for phase 21/12 and $1 - \alpha^4$ for 22/11. Note the sequence number of a cell appears in the superscript for any expected probability, and that phase number appears as subscript when there are more than two phases. The two possible phases

Table 1
Genotype counts for biallelic markers

Marker 1	Marker 2		
	1/1	1/2	2/2
1/1	n_0	n_1	n_2
1/2	n_3	n_4	n_5
2/2	n_6	n_7	n_8

Table 2
Genotypic probabilities for two biallelic markers

Marker 1	Marker 2		
	1/1	1/2	2/2
1/1	h_{11}^2	$2h_{11}h_{12}$	h_{12}^2
1/2	$2h_{21}h_{11}$	$2(h_{21}h_{12}+h_{22}h_{11})$	$2h_{22}h_{12}$
2/2	h_{21}^2	$2h_{21}h_{22}$	h_{22}^2

from cell 4 are effectively enumerated by Subroutine N. Binary numbers 00, 01 are used to indicate phases 11/22 and 12/21, respectively. A bit 1 in the second number specifies a switch in phase. Output from Subroutines M and H again is trivial; they would indicate the two phases contribute to elements 1/4, 2/3 in the haplotype frequency array, for example.

The expected haplotype counts are then obtained as follows:

$$c_{11} = 2n_0 + n_1 + n_3 + (1 - \alpha^4)n_4$$

$$c_{12} = n_1 + 2n_2 + n_5 + \alpha^4 n_4$$

$$c_{21} = n_3 + 2n_6 + n_7 + \alpha^4 n_4$$

$$c_{22} = n_5 + n_7 + 2n_8 + (1 - \alpha^4)n_4$$

Our haplotype frequency estimates are $c_{ij}/(2N)$, $i, j = 1, 2$, $N = \Sigma_{i=0}^{8} n_i$, giving probabilities g_i, $i = 0, \ldots, 8$ in Table 2.

A log-likelihood $l = \Sigma_{i=0}^{8} n_i \ln(g_i)$ can be constructed from multinomial distribution. If we denote log-likelihood obtained from assuming independence as l_0, that from the gene counting procedure as l_1, then $2(l_1 - l_0)$ asymptotically has χ^2 distribution with $(2 \times 2 - 1) - [(2 - 1) + (2 - 1)] = 1$ degree of freedom. The log-likelihood ratio test provides a test of allelic association between marker loci. Haplotype frequencies during initialisation amounts to assuming no association, i.e. statistically independent, whereas the estimated coupling information incorporates information about allelic association.

3.2. Three biallelic markers

Similarly data for three biallelic markers can be organised into Table 3.

With slightly more algebra the haplotype counts are obtained as follows

$$\begin{aligned}
c_{111} &= 2n_0 + n_1 + n_3 + \alpha^4 n_4 + n_9 + (1 - \alpha^{10})n_{10} \\
&\quad + (1 - \alpha^{12})n_{12} + \alpha_4^{13} n_{13} \\
c_{112} &= n_1 + 2n_2 + n_5 + (1 - \alpha^4)n_4 + \alpha^{10} n_{10} + n_{11} \\
&\quad + \alpha_3^{13} n_{13} + (1 - \alpha^{14})n_{14} \\
c_{121} &= n_3 + 2n_6 + n_7 + (1 - \alpha^4)n_4 + \alpha^{12} n_{12} + \alpha_2^{13} n_{13} \\
&\quad + n_{15} + (1 - \alpha^{16})n_{16} \\
c_{122} &= n_5 + n_7 + 2n_8 + \alpha^4 n_4 + \alpha_1^{13} n_{13} + \alpha^{14} n_{14} \\
&\quad + \alpha^{16} n_{16} + n_{17} \\
c_{211} &= n_9 + \alpha^{10} n_{10} + \alpha^{12} n_{12} + \alpha_1^{13} n_{13} + 2n_{18} + n_{19} \\
&\quad + n_{21} + \alpha^{22} n_{22} \\
c_{212} &= (1 - \alpha^{10})n_{10} + n_{11} + \alpha_2^{13} n_{13} + \alpha^{14} n_{14} + n_{19} \\
&\quad + 2n_{20} + (1 - \alpha^{22})n_{22} + n_{23} \\
c_{221} &= (1 - \alpha^{12})n_{12} + \alpha_3^{13} n_{13} + n_{15} + \alpha^{16} n_{16} + n_{21} \\
&\quad + (1 - \alpha^{22})n_{22} + 2n_{24} + n_{25} \\
c_{222} &= \alpha_4^{13} n_{13} + (1 - \alpha^{14})n_{14} + (1 - \alpha^{16})n_{16} + n_{17} \\
&\quad + \alpha^{22} n_{22} + n_{23} + n_{25} + 2n_{26}
\end{aligned}$$

where

$$\alpha^4 = \frac{h_{111}h_{122}}{(h_{111}h_{122} + h_{121}h_{112})}$$

$$\alpha^{10} = \frac{h_{211}h_{112}}{(h_{211}h_{112} + h_{212}h_{111})}$$

Table 3
Genotype counts for three biallelic markers

Marker 1	Marker 2	Marker 3		
		1/1	1/2	2/2
1/1	1/1	n_0	n_1	n_2
	1/2	n_3	n_4	n_5
	2/2	n_6	n_7	n_8
1/2	1/1	n_9	n_{10}	n_{11}
	1/2	n_{12}	n_{13}	n_{14}
	2/2	n_{15}	n_{16}	n_{17}
2/2	1/1	n_{18}	n_{19}	n_{20}
	1/2	n_{21}	n_{22}	n_{23}
	2/2	n_{24}	n_{25}	n_{26}

Table 4
Possible phases of triply heterozygote (1/2, 1/2, 1/2) using Subroutines N, M and H

Alleles at each marker			Bits from Subroutine N	Haplotype index
Marker 1	Marker2	Marker3		
1/2	1/2	1/2	000	1/8
1/2	1/2	2/1	001	2/7
1/2	2/1	1/2	010	3/6
1/2	2/1	2/1	011	4/5
2/1	1/2	1/2	100	5/4
2/1	1/2	2/1	101	6/3
2/1	1/2	1/2	110	7/2
2/1	1/2	2/1	111	8/1

$$\alpha^{12} = \frac{h_{211}h_{121}}{(h_{211}h_{121} + h_{221}h_{111})}$$

$$\alpha_1^{13} = \frac{h_{211}h_{122}}{(h_{211}h_{122} + h_{212}h_{121} + h_{221}h_{112} + h_{222}h_{111})}$$

$$\alpha_2^{13} = \frac{h_{212}h_{121}}{(h_{211}h_{122} + h_{212}h_{121} + h_{221}h_{112} + h_{222}h_{111})}$$

$$\alpha_3^{13} = \frac{h_{221}h_{112}}{(h_{211}h_{122} + h_{212}h_{121} + h_{221}h_{112} + h_{222}h_{111})}$$

$$\alpha_4^{13} = \frac{h_{222}h_{111}}{(h_{211}h_{122} + h_{212}h_{121} + h_{221}h_{112} + h_{222}h_{111})}$$

$$\alpha^{14} = \frac{h_{212}h_{122}}{(h_{212}h_{122} + h_{222}h_{112})}$$

$$\alpha^{16} = \frac{h_{221}h_{122}}{(h_{221}h_{122} + h_{222}h_{121})}$$

$$\alpha^{22} = \frac{h_{211}h_{122}}{(h_{211}h_{222} + h_{221}h_{212})}$$

are the expected contributions of certain phase(s) based on haplotype frequency estimates from the last iteration. For example with cell 13 there are four possible phases. As before $h_{ijk} = c_{ijk}/(2N)$, i, $j = 1$, 2, and $N = \Sigma_{i=0}^{26} n_i$, and $N = \Sigma_{i=0}^{26} n_i$.

Now the way Subroutines N, M and H work can be seen by examining cell 13, given in Table 4.

It is clear that only four phases are unique. So that bit 0 from Subroutine N means no switch, bit 1 means switch. The last column indicates indices of haplotype array from Subroutines M and H.

The genotypic probabilities g_i, $i = 0, \ldots, 26$ are now given by Table 5.

Log-likelihoods can be calculated in a similar fashion but now the log-likelihood ratio χ^2-test has $(2 \times 2 \times 2 - 1) - [(2-1) + (2-1) + (2-1)] = 4$ degrees of freedom.

4. Discussion

The examples we have just examined are the two most likely scenarios that all possible multilocus genotypes are observed. All multilocus genotypes are then iterated. Algorithm O then amounts to the following algorithm.

4.1. Algorithm A (gene counting using all possible multilocus genotypes)

1. From the raw data tally multilocus genotypes into a multidimensional contingency table.
2. For each cell in this table calculate its contribution to the haplotype counts from all phases and obtain appropriate haplotype counts and frequencies.
3. Iterate step 2 till the haplotype frequencies become stable by some prespecified criteria.

With the same logic for keeping haplotype frequencies, the implementation requires storing the contingency table as an array. Subroutine H will then be used to index the multilocus genotype array. Even with small problems this brute-force method may not be better than Algorithm O in time complexity, but the exposition is simpler.

We have not seen any source program for similar problems in the literature and searches through the Internet, and some remarks are worthwhile here.

The data preparation is a multikey sorting problem available in data management and analysis packages. A perhaps more familiar example is as follows: if our data *testfile* contains variables sex, agegroup and we are interested in obtaining information such as number of individuals for certain sex and age group, their average heights and weights. A typical SAS program would be as follows.

```
PROC SORT DATA = testfile;
  BY sex agegroup;
RUN;
PROC FREQ;
  TABLE sex*agegroup/list;
RUN;
PROC SUMMARY;
  BY sex agegroup;
  VAR weights heights;
  OUTPUT OUT = rlt N = nw nh MEAN =
  mw mh;
RUN;
```

The raw data in file is sorted by sex and agegroup first, patterns and summary statistics are then obtained.

Radix sort works well when each class has limited number of values [10]. Our implementation of radix sorting needs extra amount of storage, which is a modest drawback compared with the number of haplotype frequencies. In fact a feature of Algorithm O is that we can keep track of haplotypes appeared in the data so that the size of haplotype array is greatly reduced.

Elegant implementation of radix sorting is an open problem in general computing [11]. When the number of keys become large, we expect records in our file are more likely to be unique, so that radix sort hybrided with simple sorting would perform well. The use of mixed-radix number is perhaps more dominant in string matching problem [12–14]. To search for a pattern $P[1 \ldots m]$ of length m in a text $T[s+1 \ldots s+m]$ of length n, $s=1, \ldots, n-m$, we can use decimal values in both the pattern and text, to be calculated using Horner's method. When their values become large modular scheme such as Rabin–Karp method can be used to resolve spurious hits. Method related to DNA sequence searching was given by Karlin et al. [15,16]. Since the compositions of DNA, A, T, C, G can be 0, 1, 2, 3 in a quaternary number system.

Although we use SNPs markers as examples our algorithms also apply to other polymorphic markers such as restricted fragment-length polymorphism (RFLP) and microsatellites. The ability to handle large number of loci is important. As of March 2001, 2.84 million SNPs have been deposited in the public database, dbSNP, at the National Centre for Biotechnology Information, http://www.ncbi.nlm.nih.gov/SNP [17]. Examining patterns of their interdependence is the subject

Table 5
The genotypic probabilities for three biallelic markers

Marker 1	Marker 2	Marker 3		
		1/1	1/2	2/2
1/1	1/1	h_{111}^2	$2h_{112}h_{111}$	h_{112}^2
	1/2	$2h_{111}h_{121}$	$2h_{112}h_{121}+2h_{111}h_{122}$	$2h_{112}h_{122}$
	2/2	h_{121}^2	$2h_{121}h_{122}$	h_{122}^2
1/2	1/1	$2h_{211}h_{111}$	$2h_{211}h_{112}+2h_{212}h_{\mathrm{d}11}$	$2h_{212}h_{112}$
	1/2	$2h_{211}h_{121}+2h_{221}h_{111}$	$2h_{211}h_{122}+2h_{212}h_{\mathrm{d}21}+2h_{221}h_{112}+2h_{222}h_{111}$	$2h_{212}h_{122}$ $+2h_{222}h_{112}$
	2/2	$2h_{221}h_{121}$	$2h_{221}h_{122}+2h_{222}h_{121}$	$2h_{222}h_{122}$
2/2	1/1	h_{211}^2	$h_{211}h_{212}$	h_{212}^2
	1/2	$2h_{211}h_{221}$	$2h_{211}h_{222}+2h_{221}h_{212}$	$2h_{212}h_{222}$
	2/2	h_{221}^2	$2h_{221}h_{222}$	h_{222}^2

of association analysis, which has become backbone of fine mapping disease genes. This will have important implications on evolution and disease mutation detection. Many factors can lead to association, e.g. the marker locus examined being the disease locus itself, the disease locus in linkage disequilibrium with the marker locus, or population substructure. Allelic association using carefully matched cases and controls in a homogeneous population is a powerful and feasible strategy, and one of the major strategies exploring the genetic mechanism underlying common diseases [18,19]. Although alternative methods such as Markov chain Monte Carlo (MCMC) have been proposed [20,21], the golden standard would be actual molecular experiment [21].

There are many challenging problems in bioinformatics. Efficient solutions to such problems rest on theoretically sound and practically feasible methods, often a result of bridging gap in biology, mathematics and computing. We believe that both the generic nature of the proposed algorithms and the problems they approach deserve more attention.

Acknowledgements

We thank the Editor and an anonymous referee for editorial help and invaluable suggestions, which significantly improve the manuscript. This research is supported by Wellcome Trust project grant 055379 and HEFCE funding from King's College London.

Appendix A

Some definitions in statistical genetics and computing are briefly reviewed here and may be skipped if you are already familiar with them. More details are available from [12,22] and references therein.

Some definitions

Gene. A piece of DNA with certain function as unit of heredity.

Marker. A piece of DNA within unknown function, can be interpreted as a milestone laid on chromosomes.

Locus. A place in chromosome where genes are located.

Allele. One of the variant forms of a gene. A diploid individual has two alleles for each gene, one on each of the homologous chromosomes, at the appropriate locus for that gene. Their relative frequencies vary between different populations. If frequency of the most common allele occurss at most 99%, the locus is called polymorphic. If the two alleles are identical the individual is homozygote at that locus, and if they are different the individual is heterozygote. A single nucleotide polymorphism (SNP) has two alleles.

Genotype. The alleles present at one or more loci. For a marker locus we usually assume the two alleles are codominant, i.e. each allele expresses its effect regardless the other.

Genotype identifier. A single number to represent genotype, can be calculated as $L+U(U-1)/2$, where L and U are the low and high alleles at a locus. Multilocus identifier can be obtained similarly.

Haplotype. A set of alleles from one of the two alleles at each locus.

Linked-list. An abstract data structure created at computer run-time. Essentially each item in the list contains a key and pointer(s) to other item(s) in the list. Without explicit specification, satellite information is always bundled with keys.

Graph. Not unanimously defined in the literature, is a data structure consisting of set of points called vertices and lines connecting them called edges.

Trees. Connected, acyclic, undirected graph, here as an abstract data structure for sorting and searching.

Radix. The base of a number system. For instance [11] the decimal number $365=3\times 10^2+6\times 10+5$. Sometimes it is necessary to have digits from different bases. For instance 10-04-2001 means 10th of April, year 2001. Ranges for day, month and year can all suitably be defined.

Mixed radix sorting. A radix sorting method using mixed-radix number. For example our date key has a C structure as follows:

```
typedef struct t_date {
int day;
int month;
int year;
} date;
```

In standard radix sort, the key to each data item in a list to be sorted is first expressed in binary format, and then sorted according to a bit is 0 or 1 [10], or a bunch of bits to a base of 4, 8, 16, etc. Here the same sorting procedure is applied except that a different number of bins are employed in each phase. Either the least significant digit or most significant digit (where 'digit' here means a field with a limited range) can be used.

Horner's method. An efficient way of polynomial evaluation in which a nested calculation is used instead of using direct exponentiation. For example $365 = (3 \times 10 + 6)10 + 5$.

References

[1] J.H. Zhao, P.C. Sham, A method for calculating probability convolution using ternary numbers with application in the determination of twin zygosity, Comp. Stat. Data Anal. 28 (1998) 225–232.

[2] P.C. Sham, J.H. Zhao, D. Curtis, The effect of marker characteristics on the power to detect linkage disequilibrium due to single or multiple ancestral mutations, Ann. Hum. Genet. 64 (2000) 161–169.

[3] R. Ceppellini, M. Siniscalco, C.A.B. Smith, The estimation of gene frequencies in a random mating population, Ann. Hum. Genet. 20 (1955) 97–115.

[4] W.G. Hill, Tests for association of gene frequencies at several loci in random mating diploid populations, Biometrics 31 (1975) 881–888.

[5] J. Ott, Genet couning methods (EM algorithm) in human pedigree analysis: linkage and segregation analysis, Ann. Hum. Genet. 40 (1977) 443–454.

[6] J.H. Zhao, D. Curtis, P.C. Sham, Model-free analysis and permutation tests for allelic associations, Hum. Hered. 50 (2000) 133–139.

[7] G.H. Gonnet, R. Baeza-Yates, Handbook of Algorithms and Data Structures. (Addison-Wesley, http://www.dcc.uchile.cl/ ~ rbaeza/handbook/hbook.html, 1991).

[8] J.D. Terwilliger, J. Ott, Handbook of Human Genetic Linkage, The Johns Hopkins University Press, Baltimore, 1994.

[9] P.C. Sham, Statistics in Human Genetics, Edward Arnold, London, 1998.

[10] D.E. Knuth, The Art of Computer Programming. Sorting and Search, vol. 3, Addison-Wesley, Reading, MA, 1998.

[11] J. Morris, Data structure and algorithms. (http://swww.ee.uwa.edu.au/ ~ plsd210/ds/, 1998).

[12] T.H. Corman, C.E. Leisrson, R.L. Rviest, Introduction to Algorithms, MIT Press, Cambridge, MA, 1990.

[13] S. Nilsson, Radix sorting and searching. Ph.D. Thesis, Department of Computer Science, Lund University, 1996.

[14] J.L. Bentley, R. Sedgewick, Fast algorithms for sorting and searching strings. (http://www.cs.princeton.edu).

[15] S. Karlin, S.F. Altschul, Methods for assessing the significance of molecular sequence features by using general scoring schemes, Proc. Natl. Acad. Sci. USA 87 (1990) 2264–2268.

[16] B.S. Weir, Genetic Data Analysis II, Sinauer Associates, Sunderland, MA, 1996.

[17] G. Marth, R. Yeh, M. Minton, R. Donaldson, Q. Li, S. Duan, R. Davenport, R.D. Miller, P.-Y. Kwok, Single-nucleotide polymorphisms in the public domain: how useful are they, Nat. Genet. 27 (2001) 371–372.

[18] B. Keavney, C. McKenzie, S. Parish, A. Palmer, S. Clark, L. Youngman, M. Delèpine, et al., Large-scale test of hypothesised associations between the angiotensin-converting-enzyme insertion/deletion polymorphism and myocardial infarction in about 5000 cases and 6000 controls, Lancet 355 (2000) 434–442.

[19] H. Zhao, S. Zhang, K.R. Merikangas, M. Trixler, D.B. Wildenauer, F. Sun, K.K. Kidd, Transmission/disequilibrium tests using multiple tightly linked markers, Am. J. Hum. Genet. 67 (2000) 936–946.

[20] L.C. Lazzeroni, K. Lange, Markov chains for Monte Carlo tests of genetic equilibrium in multidimensional contingency tables, Ann. Stat. 25 (1997) 138–168.

[21] M. Stephens, N.J. Smith, P. Donnelly, A new statistical method for haplotype reconstruction from population data, Am. J. Hum. Genet. 68 (2001) 978–989.

[22] R.C. Elston, Introduction and overview, Stat. Methods Med. Res. 9 (2000) 527–541.